

# Learning where to Attend with Deep Architectures for Image Tracking

**Misha Denil<sup>1</sup>, Loris Bazzani<sup>2</sup>, Hugo Larochelle<sup>3</sup> and Nando de Freitas<sup>1</sup>**

<sup>1</sup>University of British Columbia.

<sup>2</sup>University of Verona.

<sup>3</sup>University of Sherbrooke.

**Keywords:** Restricted Boltzmann machines, Bayesian optimization, bandits, attention, deep learning, particle filtering, saliency

## Abstract

We discuss an attentional model for simultaneous object tracking and recognition that is driven by gaze data. Motivated by theories of perception, the model consists of two interacting pathways: identity and control, intended to mirror the what and where pathways in neuroscience models. The identity pathway models object appearance and performs classification using deep (factored)-Restricted Boltzmann Machines. At each point in time the observations consist of foveated images, with decaying resolution toward the periphery of the gaze. The control pathway models the location, orientation, scale and speed of the attended object. The posterior distribution of these states is estimated with particle filtering. Deeper in the control pathway, we encounter an attentional mechanism that learns to select gazes so as to minimize tracking uncertainty. Unlike in our previous work, we introduce gaze selection strategies which operate in the presence

of partial information and on a continuous action space. We show that a straightforward extension of the existing approach to the partial information setting results in poor performance, and we propose an alternative method based on modeling the reward surface as a Gaussian Process. This approach gives good performance in the presence of partial information and allows us to expand the action space from a small, discrete set of fixation points to a continuous domain.

## 1 Introduction

Humans track and recognize objects effortlessly and efficiently, exploiting attentional mechanisms (Rensink, 2000; Colombo, 2001) to cope with the vast stream of data. We use the human visual system as inspiration to build a system for simultaneous object tracking and recognition from gaze data. An attentional strategy is learned online to choose fixation points which lead to low uncertainty in the location of the target object. Our tracking system is composed of two interacting pathways. This separation of responsibility is a common feature in models from the computational neuroscience literature as it is believed to reflect a separation of information processing into ventral and dorsal pathways in the human brain (Olshausen et al., 1993a).

The *identity* pathway (ventral) is responsible for comparing observations of the scene to an object template using an appearance model, and on a higher level, for classifying the target object. The identity pathway consists of a two hidden layer deep network. The top layer corresponds to a multi-fixation Restricted Boltzmann Machine (RBM) (Larochelle & Hinton, 2010), as shown in Figure 1. It accumulates information from the first hidden layers at consecutive time steps. For the first layers, we use (factored)-RBMs (Hinton & Salakhutdinov, 2006; Ranzato & Hinton, 2010; Welling et al., 2005; Swersky et al., 2011), but autoencoders (Vincent et al., 2008), sparse coding (Olshausen & Field, 1996; Kavukcuoglu et al., 2009), two-layer ICA (Köster & Hyvärinen, 2007) and convolutional architectures (Lee et al., 2009) could also be adopted.

The *control* pathway (dorsal) is responsible for aligning the object template with the full scene so the remaining modules can operate independently of the object’s position and scale. This pathway is separated into a localization module and a fixation module

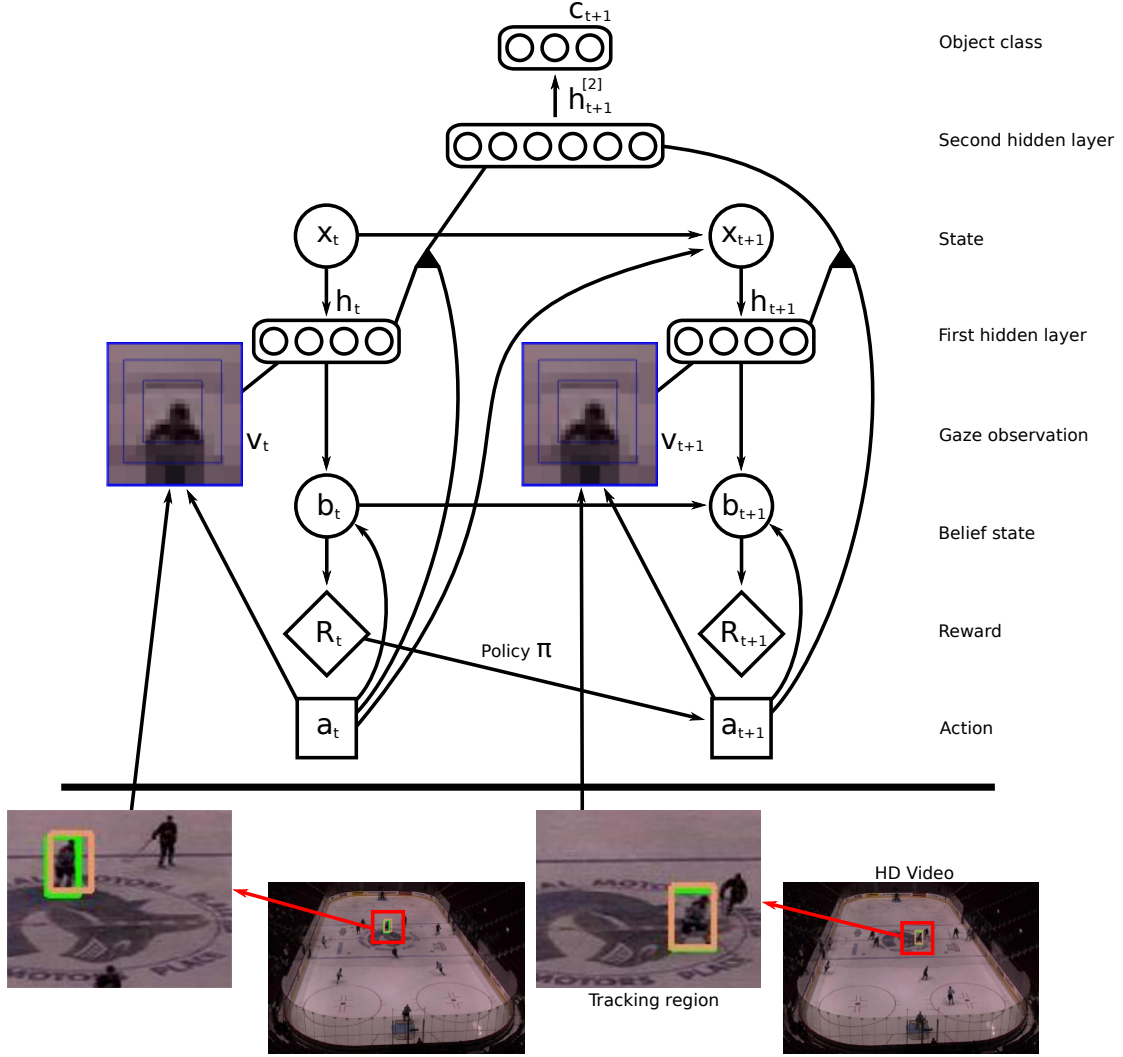


Figure 1: From a sequence of gazes ( $\mathbf{v}_t, \mathbf{v}_{t+1}, \dots$ ), the model infers the hidden features  $\mathbf{h}$  for each gaze (that is, the activation intensity of each receptive field), the hidden features for the fusion of the sequence of gazes and the object class  $\mathbf{c}$ . Only one time step of classification is kept in the figure for clarity. The location, size, speed and orientation of the gaze patch are encoded in the state  $\mathbf{x}_t$ . The actions  $\mathbf{a}_t$  follow a learned policy  $\pi_t$  that depends on the past rewards  $\{r_1, \dots, r_{t-1}\}$ . This particular reward is a function of the belief state  $\mathbf{b}_t = p(\mathbf{x}_t | \mathbf{a}_{1:t}, \mathbf{h}_{1:t})$ , also known as the filtering distribution. Unlike typical commonly used partially observed Markov decision models (POMDPs), the reward is a function of the beliefs. In this sense, the problem is closer to one of sequential experimental design. With more layers in the ventral  $\mathbf{v} - \mathbf{h} - \mathbf{h}^{[2]} - \mathbf{c}$  pathway, other rewards and policies could be designed to implement higher-level attentional strategies.

which work cooperatively to accomplish this goal. The localization module is implemented with a particle filter (Doucet et al., 2001) which estimates the location, velocity and scale of the target object. We make no attempt to implement such states with neural architectures, but it seems clear that they could be encoded with grid cells (McNaughton et al., 2006) and retinotopic maps as in V1 and the superior colliculus (Rosa, 2002; Girard & Berthoz, 2005). The fixation module learns an attentional strategy to select fixation points relative to the object template. These fixation points are the centres of partial template observations, and are compared with observations of the corresponding locations in the scene using the appearance model (see Figure 2). Reward is assigned to each fixation based on the uncertainty of the target location at each time step. The fixation module uses the reward signal to adapt its gaze selection policy to achieve good localization. Our previous work (Bazzani et al., 2010) used Hedge (Auer et al., 1998a; Freund & Schapire, 1997) to learn this policy. In this extended paper we show that a straightforward adaptation of our previous approach to the partial information setting results in poor performance, and we propose an alternative method based on modelling the reward surface as a Gaussian Process. This approach gives good performance in the presence of partial information and allows us to expand the action space from a small, discrete set of fixation points to a continuous domain.

The proposed system can be motivated from different perspectives. First, starting with Isard & Blake (1996), many particle filters have been proposed for image tracking, but these typically use simple observation models such as B-splines (Isard & Blake, 1996) and colour templates (Okuma et al., 2004). RBMs are more expressive models of shape, and hence, we conjecture that they will play a useful role where simple appearance models fail. Second, from a deep learning computational perspective, this work allows us to tackle large images and video, which is typically not possible due to the number of parameters required to represent large images in deep models. The use of fixations synchronized with information about the state (e.g. location and scale) of such fixations eliminates the need to look at the entire image or video. Third, the system is invariant to image transformations encoded in the state, such as location, scale and orientation. Fourth, from a dynamic sensor network perspective, this paper presents a very simple, but efficient and novel way of deciding how to gather measurements dynamically. Lastly, in the context of psychology, the proposed model realizes

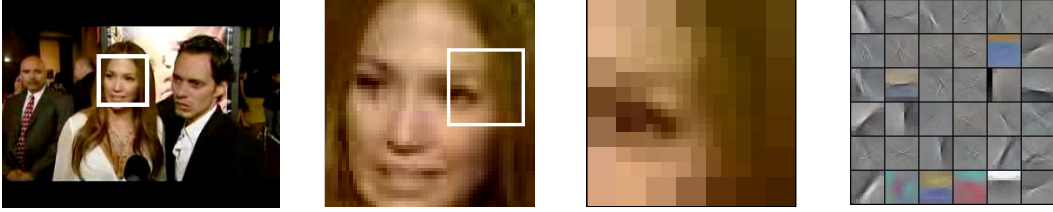


Figure 2: **Left:** A typical video frame with the estimated target region highlighted. To cope with the large image size our system considers only the target region at each time step. **Centre left:** A close-up of the template extracted from the first frame. The template is compared to the target region by selecting a fixation point for comparison as shown. **Centre right:** A visualization of a single fixation. In addition to covering only a very small portion of the original frame, the image is foveated with high resolution near the centre and low resolution on the periphery to further reduce the dimensionality. **Right:** The most active filters of the first layer (factored)-RBM when observing the displayed location. The control pathway compares these features to the features active at the corresponding scene location in order to update the belief state.

to some extent the functional architecture for dynamic scene representation of Rensink (2000). The rate at which different attentional mechanisms develop in newborns (including alertness, saccades and smooth pursuit, attention to object features and high-level task driven attention) guided the design of the proposed approach and was a great source of inspiration (Colombo, 2001).

Our attentional model can be seen as building a saliency map (Koch & Ullman, 1985) over the target template. Previous work on saliency modelling has focused on identifying salient points in an image using a bottom up process which looks for outliers under some local feature model (which may include a task dependent prior, global scene features, or various other heuristics). These features can be computed from static images (Torralba et al., 2006), or from local regions of spacetime (Gaborski et al., 2004) for video. Additionally, a wide variety of different feature types have been applied to this problem, including engineered features (Gao et al., 2007) as well as features that are learned from data (Zhang et al., 2009). Core to these methods is the idea that saliency is determined by some type of novelty measure. Our approach is different, in that rather than identifying locally or globally novel features, our process identifies features which are useful for the task at hand. In our system the saliency signal for a location comes from a top down process which evaluates how well the features at that location enable

the system to localize the target object. The work of Gao et al. (2007) considers a similar approach to saliency by defining saliency to be the mutual information between the features at a location and the class label of an object being sought; however, in order to make their model tractable the authors are forced to use specifically engineered features. Our system is able to cope with arbitrary feature types, and although we consider only on localization in this paper, our model is sufficiently general to be applied to identifying salient features for other goals.

Recently, a dynamic RBM state-space model was proposed in Taylor et al. (2010). Both the implementation and intention behind that proposal are different from the approach discussed here. To the best of our knowledge, our approach is the first successful attempt to combine dynamic state estimation from gazes with online policy learning for gaze adaptation, using deep network models of appearance. Many other dual-pathway architectures have been proposed in computational neuroscience, including Olshausen et al. (1993b) and Postma et al. (1997), but we believe ours has the advantage that it is very simple, *modular* (with each module easily replaceable), suitable for large datasets and easy to extend.

## 2 Identity Pathway

The identity pathway in our model mirrors the ventral pathway in neuroscience models. It is responsible for modelling the appearance of the target object and also, at a higher level, for classification.

### 2.1 Appearance Model

We use (factored)-RBMs to model the appearance of objects and perform object classification using the gazes chosen by the control module (see Figure 3). These undirected probabilistic graphical models are governed by a Boltzmann distribution over the gaze data  $\mathbf{v}_t$  and the hidden features  $\mathbf{h}_t \in \{0, 1\}^{n_h}$ . We assume that the receptive fields  $\mathbf{w}$ , also known as RBM weights or filters, have been trained beforehand. We also assume that readers are familiar with these models and, if otherwise, refer them to Ranzato & Hinton (2010) and Swersky et al. (2010).

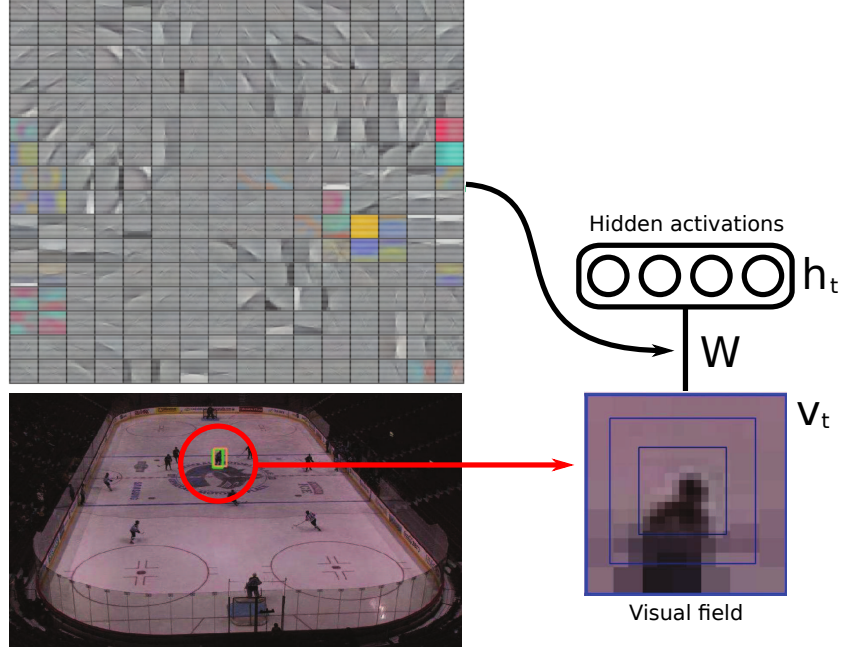


Figure 3: An RBM senses a small foveated image derived from the video. The level of activation of each filter is recorded in the  $\mathbf{h}_t$  units. The RBM weights (filters)  $\mathbf{W}$  are visualized in the upper left. We currently pre-train these weights.

## 2.2 Classification Model

The identity pathway also performs object recognition, classifying a sequence of gaze instances selected with the gaze policy. We implement a multi-fixation RBM very similar to the one proposed in Larochelle & Hinton (2010), where the binary variables  $\mathbf{z}_t$  (see Figure 4) are introduced to encode the relative gaze location  $\mathbf{a}_t$  within the multi-fixation RBM (a “1 in  $K$ ” or “one hot” encoding of the gaze location was used for  $\mathbf{z}_t$ ).

The multi-fixation RBM uses the relative gaze location information in order to aggregate the first hidden layer representations  $\mathbf{h}_t$  at  $\Delta$  consecutive time steps into a single, higher level representation  $\mathbf{h}_t^{[2]}$ .

More specifically, the energy function of the multi-fixation RBM is:

$$E(\mathbf{h}_{t-\Delta+1:t}, \mathbf{z}_{t-\Delta+1:t}, \mathbf{h}_t^{[2]}) = -\mathbf{d}^\top \mathbf{h}_t^{[2]} - \sum_{i=1}^{\Delta} \mathbf{b}^\top \mathbf{h}_{t-\Delta+i} + \sum_{f=1}^F (\mathbf{P}_{f,:} \mathbf{h}_t^{[2]})(\mathbf{W}_{f,:} \mathbf{h}_{t-\Delta+i})(\mathbf{V}_{f,:} \mathbf{z}_{t-\Delta+i})$$

where the notation  $\mathbf{P}_{f,:}$  refers to the  $f^{\text{th}}$  row vector of the matrix  $\mathbf{P}$ . From this energy function, we define a distribution over  $\mathbf{h}_{t-\Delta+1:t}$  and  $\mathbf{h}_t^{[2]}$  (conditioned on  $\mathbf{z}_{t-\Delta+1:t}$ )

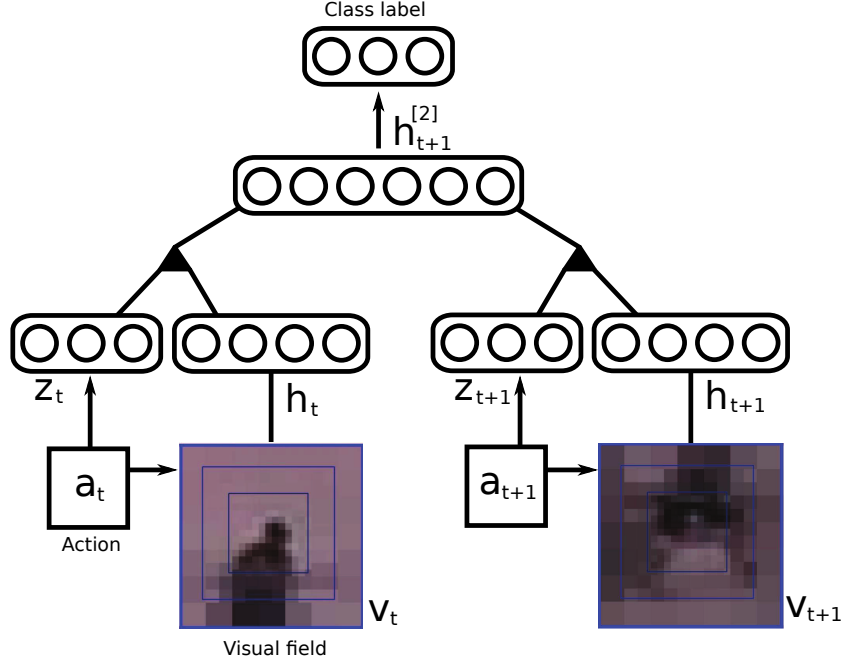


Figure 4: *Gaze accumulation and classification in the identity pathway. A multi-fixation RBM models the conditional distribution (given the gaze positions  $\mathbf{a}_t$ ) of  $\Delta$  consecutive hidden features  $\mathbf{h}_t$ , extracted by the first layer RBM on the foveated images. In this illustration,  $\Delta = 2$ . The multi-fixation RBM encodes the gaze position  $\mathbf{a}_t$  in a “one hot” representation noted  $\mathbf{z}_t$ . The activation probabilities of the second layer hidden units  $\mathbf{h}_t^{[2]}$  are used by a classifier to predict the object’s class.*

through the Boltzmann distribution:

$$p(\mathbf{h}_{t-\Delta+1:t}, \mathbf{h}_t^{[2]} | \mathbf{z}_{t-\Delta+1:t}) = \exp \left( -E(\mathbf{h}_{t-\Delta+1:t}, \mathbf{z}_{t-\Delta+1:t}, \mathbf{h}_t^{[2]}) \right) / Z(\mathbf{z}_{t-\Delta+1:t}) \quad (1)$$

where the normalization constant  $Z(\mathbf{z}_{t-\Delta+1:t})$  ensures that Equation 1 sums to 1. To sample from this distribution, one can use Gibbs sampling by alternating between sampling the top-most hidden layer  $\mathbf{h}_t^{[2]}$  given all individual processed gazes  $\mathbf{h}_{t-\Delta+1:t}$  and vice versa. To train the multi-fixation RBM, we collect a training set consisting in sequences of  $\Delta$  pairs  $(\mathbf{h}_t, \mathbf{z}_t)$  by randomly selecting  $\Delta$  gaze positions at which to fixate and computing the associated  $\mathbf{h}_t$ . These sets are extracted from a collection of images in which the object to detect has been centred. Unsupervised learning using contrastive divergence can then be performed on this training set. See Larochelle & Hinton (2010) for more details.

The main difference between this multi-fixation RBM and the one described in



Larochelle & Hinton (2010) is that  $\mathbf{h}_t^{[2]}$  does not explicitly model the class label  $\mathbf{c}_t$ . Instead, a multinomial logistic regression classifier is trained separately, to predict  $\mathbf{c}_t$  from the aggregated representation extracted from  $\mathbf{h}_t^{[2]}$ . More specifically, we use the vector of activation probabilities of all hidden units  $h_{t,j}^{[2]}$  in  $\mathbf{h}_t^{[2]}$ , conditioned on  $\mathbf{h}_{t-\Delta+1:t}$  and  $\mathbf{z}_{t-\Delta+1:t}$ , as the aggregated representation:

$$p(h_{t,j}^{[2]} = 1 | \mathbf{h}_{t-\Delta+1:t}, \mathbf{z}_{t-\Delta+1:t}) = \text{sigm} \left( d_j + \sum_{i=1}^{\Delta} \sum_{f=1}^F \mathbf{P}_{f,j}(\mathbf{W}_{f,:} \mathbf{h}_{t-\Delta+i})(\mathbf{V}_{f,:} \mathbf{z}_{t-\Delta+i}) \right)$$

We experimented with a single fixation module, but found the multi-fixation module to increase classification accuracy. To improve the estimate the class variable  $\mathbf{c}_t$  over time, we accumulate the classification decisions at each time step.

Note that the process of pursuit (tracking) is essential to classification. As the target is tracked, the algorithm fixates at locations near the target’s estimated location. The size and orientation of these fixations also depends on the corresponding state estimates. Note that we don’t fixate exactly at the target location estimate as this would provide only one distinct fixation over several time steps if the tracking policy has converged to a specific gaze. It should also be pointed out that instead of using random fixations, one could again use the control strategy proposed in this paper to decide where to look with respect to the track estimate so as to reduce classification uncertainty. We leave the implementation of this extra attentional mechanism for future work.

### 3 Control Pathway

The control pathway mirrors the responsibility of the dorsal pathway in human visual processing. It tracks the state of the target (position, speed, etc) and normalizes the input so that other modules need not account for these variations. At a higher level it is responsible for learning an attentional strategy which maximizes the amount of information learned with each fixation. The structure of the control pathway is shown in Figure 5.

#### 3.1 State-space model

The standard approach to image tracking is based on the formulation of Markovian, nonlinear, non-Gaussian state-space models, which are solved with approximate Bayesian

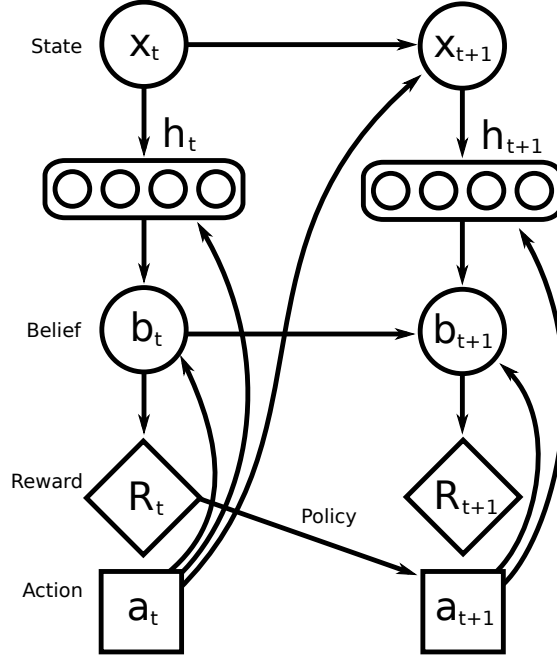


Figure 5: Influence diagram for the control pathway. The true state of the tracked object  $\mathbf{x}_t$ , generates some set of features  $\mathbf{h}_t$ , in the identity pathway. These features depend on the action chosen at time  $t$  and are used to update the belief state  $\mathbf{b}_t$ . Statistics of the belief state are collected to compute the reward  $r_t$ , which is used to update the policy for the next time step.

filtering techniques. In this setting, the unobserved signal (object’s position, velocity, scale, orientation or discrete set of operations) is denoted  $\{\mathbf{x}_t \in \mathcal{X}; t \in \mathbb{N}\}$ . This signal has initial distribution  $p(\mathbf{x}_0)$  and transition equation  $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{a}_{t-1})$ . Here  $\mathbf{a}_t \in \mathcal{A}$  denotes an action at time  $t$ , defined on a compact set  $\mathcal{A}$ . For discrete policies  $\mathcal{A}$  is finite whereas for continuous policies  $\mathcal{A}$  is a region in  $\mathbb{R}^2$ . The observations  $\{\mathbf{h}_t \in \mathcal{H}; t \in \mathbb{N}^*\}$ , are assumed to be conditionally independent given the process state  $\{\mathbf{x}_t; t \in \mathbb{N}\}$ . Note that from the state space model perspective the observations are the hidden units of the second layer of the appearance model in the identity pathway. In summary, the state-space model is described by the following distributions:

$$\begin{aligned}
 & p(\mathbf{x}_0) \\
 & p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{a}_{t-1}) \quad \text{for } t \geq 1 \\
 & p(\mathbf{h}_t | \mathbf{x}_t, \mathbf{a}_t) \quad \text{for } t \geq 1,
 \end{aligned}$$

For the transition model, we will adopt a classical autoregressive process.

Our aim is to estimate recursively in time the *posterior distribution*<sup>1</sup>  $p(\mathbf{x}_{0:t} | \mathbf{h}_{1:t}, \mathbf{a}_{1:t})$  and its associated features, including the marginal distribution  $\mathbf{b}_t \triangleq p(\mathbf{x}_t | \mathbf{h}_{1:t}, \mathbf{a}_{1:t})$  — known as the *filtering distribution* or *belief state*. This distribution satisfies the following recurrence:

$$\mathbf{b}_t \propto p(\mathbf{h}_t | \mathbf{x}_t, \mathbf{a}_t) \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{a}_{t-1}) p(d\mathbf{x}_{t-1} | \mathbf{h}_{1:t-1}, \mathbf{a}_{1:t-1}).$$

Except for standard distributions (e.g. Gaussian or discrete), this recurrence is intractable.

After learning the observation model we will use it for tracking. The observation model is often defined in terms of the distance of the observations from a template  $\tau$ ,

$$p(\mathbf{h}_t | \mathbf{x}_t, \mathbf{a}_t) \propto \exp(-d(\mathbf{h}(\mathbf{x}_t, \mathbf{a}_t), \tau)),$$

where  $d(\cdot, \cdot)$  denotes a distance metric and  $\tau$  an object template (for example, a color histogram or spline). In this model, the observation  $\mathbf{h}(\mathbf{x}_t, \mathbf{a}_t)$  is a function of the current state hypothesis and the selected action. The problem with this approach is eliciting a good template. Often color histograms or splines are insufficient. For this reason, we will construct the templates with (factored)-RBMs as follows. First, optical flow is used to detect new object candidates entering the visual scene. Second, we assign a template to the detected object candidate, as shown in Figure 2. The same figure also shows a typical foveated observation (higher resolution in the center and lower in the periphery of the gaze) and the receptive fields for this observation learned beforehand with an RBM. The control algorithm will be used to learn which parts of the template are most informative, either by picking from among a predefined set of fixation points, or by using a continuous policy. Finally, we define the likelihood of each observation directly in terms of the distance of the hidden units of the RBM  $\mathbf{h}(\mathbf{x}_t, \mathbf{a}_t, \mathbf{v}_t)$ , to the hidden units of the corresponding template region  $\mathbf{h}(\mathbf{x}_1, \mathbf{a}_t = k, \mathbf{v}_1)$ . That is,

$$p(\mathbf{h}_t | \mathbf{x}_t, \mathbf{a}_t = k) \propto \exp(-d(\mathbf{h}(\mathbf{x}_t, \mathbf{a}_t = k, \mathbf{v}_t), \mathbf{h}(\mathbf{x}_1, \mathbf{a}_t = k, \mathbf{v}_1))).$$

The above template is static, but conceivably one could adapt it over time.

### 3.2 Reward Function

A gaze control strategy specifies a policy  $\pi(\cdot)$  for selecting fixation points. The purpose of this strategy is to select fixation points which maximize an instantaneous reward

---

<sup>1</sup> We use the notation  $\mathbf{x}_{0:t} \triangleq \{\mathbf{x}_0, \dots, \mathbf{x}_t\}$  to represent the past history of a variable over time.

function  $r_t(\cdot)$ . The reward can be any desired behaviour for the system, such as minimizing posterior uncertainty or achieving a more abstract goal. We focus on gathering observations so as to minimize the uncertainty in the estimate of the filtering distribution:  $r_t(\mathbf{a}_t|\mathbf{b}_t) \triangleq u[\tilde{p}(\mathbf{x}_t|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})]$ . More specifically, as discussed later, this reward will be a function of the variance of the importance weights of the particle filter approximation  $\tilde{p}(\mathbf{x}_t|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})$  of the belief state.

It is also useful to consider the cumulative reward

$$R_T = \sum_{t=1}^T r_t(\mathbf{a}_t|\mathbf{b}_t) ,$$

which is the sum of the instantaneous rewards which have been received up to time  $T$ . The gaze control strategies we consider are all “no-regret” which means that the average gap between our cumulative reward and the cumulative reward from always picking the optimal action goes to zero as  $T \rightarrow \infty$ .

In our current implementation, each action is a different gaze location and the objective is to choose where to look so as to minimize the uncertainty about the belief state.

## 4 Gaze control

We compare several different strategies for learning the gaze selection policy. In an earlier version of this work (Bazzani et al., 2010) we learned the gaze selection policy with a portfolio allocation algorithm called Hedge (Freund & Schapire, 1997; Auer et al., 1998b). Hedge requires knowledge of the rewards for all actions at each time step, which is not realistic when gazes must be preformed sequentially, since the target object will move between fixations. We compare this strategy, as well as two baseline methods, to two very different alternatives.

EXP3 is an extension of Hedge to partial information games (Auer et al., 2001). Unlike Hedge, EXP3 requires knowledge of the reward only for the action selected at each time step. EXP3 is more appropriate to the setting at hand, and is also more computationally efficient than Hedge; however, this comes at a cost of substantially lower theoretical performance.

Both Hedge and EXP3 learn gaze selection policies which choose among a discrete

set of predetermined fixation points. We can instead learn a continuous policy by estimating the reward surface using a Gaussian Process (Rasmussen & Williams, 2006). By assuming that the reward surface is smooth, we can draw on the tools of Bayesian optimization (Brochu et al., 2010) to search for the optimal gaze location using as few exploratory steps as possible.

The following sections describe each of these approaches in more detail.

## 4.1 Baseline

We consider two baseline strategies, which we call random and circular. The random strategy samples gaze selections uniformly from a small discrete set of possibilities. The circular strategy also uses a small discrete set of gaze locations and cycles through them in a fixed order.

## 4.2 Hedge

To use Hedge (Freund & Schapire, 1997; Auer et al., 1998b) for gaze selection we must first discretize the action space by selecting a fixed finite number of possible fixation points. Hedge maintains an importance weight  $G(i)$  for each possible fixation point and uses them to form a stochastic policy at each time step. An action is selected according to this policy and the reward for each possible action is observed. These rewards are then used to update the importance weights and the process repeats. Pseudo code for Hedge is shown in Algorithm 1.

---

### Algorithm 1 Hedge

---

**Input:**  $\gamma > 0$

---

**Input:**  $G_0(i) \leftarrow 0$     **foreach**  $i \in \mathcal{A}$

**for**  $t = 1, 2, \dots$  **do**

**for**  $i \in \mathcal{A}$  **do**

$$p_t(i) \leftarrow \frac{\exp(\gamma G_{t-1}(i))}{\sum_{j \in \mathcal{A}} \exp(\gamma G_{t-1}(j))}$$

$\mathbf{a}_t \sim (p_t(1), \dots, p_t(|\mathcal{A}|))$     // sample an action from the distribution  $(p_t(k))$

**for**  $i \in \mathcal{A}$  **do**

$$r_t(i) \leftarrow r_t(i|\mathbf{b}_t)$$

$$G_t(i) \leftarrow G_{t-1}(i) + r_t(i)$$


---

### 4.3 EXP3

EXP3 (Auer et al., 2001) is a generalization of Hedge to the partial information setting. In order to maintain estimates for the importance weights, Hedge requires reward information for each possible action at each time step. EXP3 works by wrapping Hedge in an outer loop which simulates a fully observed reward vector at each time step. EXP3 selects actions based on a mixture of the policy found by Hedge and a uniform distribution. EXP3 is able to function in the presence of partial information, but this comes at the cost of substantially worse theoretical guarantees. Pseudo code for EXP3 is shown in Algorithm 2.

---

#### Algorithm 2 EXP3

---

**Input:**  $\gamma \in (0, 1]$

Initialize **Hedge**( $\gamma$ )

**for**  $t \in 1, 2, \dots$  **do**

    Receive  $\mathbf{p}_t$  from **Hedge**

$\hat{\mathbf{p}}_t \leftarrow (1 - \gamma)\mathbf{p}_t + \frac{\gamma}{|\mathcal{A}|}$

$\mathbf{a}_t \sim (\hat{p}_t(1), \dots, \hat{p}_t(|\mathcal{A}|))$

    Simulate reward vector for **Hedge** where  $\hat{r}_t(j) \leftarrow \begin{cases} r_t(j)/p_t(j) & \text{if } j = \mathbf{a}_t \\ 0 & \text{otherwise} \end{cases}$

---

### 4.4 Bayesian Optimization

Both Hedge and EXP3 discretize the space of possible fixation points and learn a distribution over this finite set. In contrast, Bayesian optimization is able to treat the space of possible fixation points as fully continuous by placing a smoothness prior on how reward is expected to vary with location. Intuitively, if we know the reward at one location, then we expect other, nearby locations to produce similar rewards. Gaussian Process priors encode this type of belief (Rasmussen & Williams, 2006), and have been used extensively for optimization of cost functions when it is important to minimize the total number of function evaluations (Brochu et al., 2010).

We model the latent reward function  $r_t(\mathbf{a}_t|\mathbf{b}_t) \triangleq r(\mathbf{a}_t|\mathbf{b}_t, \boldsymbol{\theta}_t)$  as a zero mean Gaus-

sian Process

$$r(\mathbf{a}_t | \mathbf{b}_t, \boldsymbol{\theta}_t) \sim \mathcal{GP}(\mathbf{0}, k(\mathbf{a}_t, \mathbf{a}'_t | \mathbf{b}_t, \boldsymbol{\theta}_t)) ,$$

where  $\mathbf{b}_t$  is the belief state (see Section 3.1), and  $\boldsymbol{\theta}_t$  are the model hyperparameters. The kernel function  $k(\cdot, \cdot)$ , gives the covariance between the reward at any two gaze locations. To ease the notation, the explicit dependence of  $r(\cdot)$  and  $k(\cdot, \cdot)$  on  $\mathbf{b}_t$  and  $\boldsymbol{\theta}_t$  will be dropped.

We assume that the true reward function  $r(\cdot)$  is not directly measurable, and what we observe are measurements of this function corrupted by Gaussian noise. That is, at each time step the instantaneous reward  $r_t$ , is given by

$$r_t = r(\mathbf{a}_t) + \sigma_n \delta_n ,$$

where  $\delta_n \sim \mathcal{N}(0, 1)$  and  $\sigma_n$  is a hyperparameter indicating the amount of observation noise, which we absorb into  $\boldsymbol{\theta}_t$ .

Given a set of observations we can compute the posterior predictive distribution for  $r(\cdot)$ :

$$\begin{aligned} r(\mathbf{a} | \mathbf{r}_{1:t}, \mathbf{a}_{1:t}) &\sim \mathcal{N}(m_t(\mathbf{a}), s_t^2(\mathbf{a})) , \\ m_t(\mathbf{a}) &= \mathbf{k}^T [\mathbf{K} + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{r}_{1:t} , \\ s_t^2(\mathbf{a}) &= k(\mathbf{a}, \mathbf{a}) - \mathbf{k}^T [\mathbf{K} + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{k} , \end{aligned} \tag{2}$$

where

$$\begin{aligned} \mathbf{K} &= \begin{bmatrix} k(\mathbf{a}_1, \mathbf{a}_1) & \cdots & k(\mathbf{a}_1, \mathbf{a}_t) \\ \vdots & \ddots & \vdots \\ k(\mathbf{a}_t, \mathbf{a}_1) & \cdots & k(\mathbf{a}_t, \mathbf{a}_t) \end{bmatrix} , \\ \mathbf{k} &= \begin{bmatrix} k(\mathbf{a}_1, \mathbf{a}) & \cdots & k(\mathbf{a}_t, \mathbf{a}) \end{bmatrix}^T , \\ \mathbf{r}_{1:t} &= \begin{bmatrix} r_1 & \cdots & r_t \end{bmatrix}^T . \end{aligned}$$

It remains to specify the form of the kernel function,  $k(\cdot, \cdot)$ . We experimented with several possibilities, but found that the specific form of the kernel function is not critical to the performance of this method. For the experiments in this paper we used the squared exponential kernel,

$$k(\mathbf{a}_i, \mathbf{a}_j) = \sigma_m^2 \exp \left( -\frac{1}{2} \sum_{k=1}^D \left( \frac{a_{i,k} - a_{j,k}}{\ell_k} \right)^2 \right) ,$$

where  $\sigma_m^2$  and the  $\{\ell_1, \dots, \ell_D\}$  are hyperparameters.

Equation 2 is a Gaussian Process estimate of the reward surface and can be used to select a fixation point for the next time step. This estimate gives both a predicted reward value and an associated uncertainty for each possible fixation point. This is the strength of Gaussian Processes for this type of optimization problem, since the predictions can be used to balance exploration (choosing a fixation point where the reward is highly uncertain) and exploitation (choosing a point we are confident will have high reward).

There are many selection methods available in the literature which offer different tradeoffs between these two criteria. In this paper we use GP-UCB (Srinivas et al., 2010) which selects

$$\mathbf{a}_{t+1} = \arg \max_{\mathbf{a}} m_t(\mathbf{a}) + \sqrt{\beta_t s_t(\mathbf{a})} \quad (3)$$

where  $\beta_t$  is a parameter. The setting  $\beta_t = 2 \log(t^3 \pi^2 / 3\delta)$  (with  $\delta = 0.001$ ) is used throughout this paper.

Equation 3 must still be optimized to find  $\mathbf{a}_{t+1}$ , which can be performed using standard global optimization tools. We use DIRECT (Jones et al., 1993) due to the existence of a readily available implementation.

The Gaussian Process regression is controlled by several hyperparameters (see Figure 6):  $\sigma_m^2$  controls the overall magnitude of the covariance, and  $\sigma_n^2$  controls the amount of observation noise. The remaining parameters  $\{\ell_1, \dots, \ell_D\}$  are length scale parameters which control the range of the covariance effects in each dimension.

Treatment of the hyperparameters requires special consideration in this setting. The pure Bayesian approach is to put a prior on each parameter and integrate them out of the predictive distribution. However, since the integrals involved are not tractable analytically, this requires computationally expensive numerical approximations. Speed is an issue here since GP-UCB requires that we optimize a function of the posterior process at each time step so, for instance, computing Monte Carlo averages for each evaluation of Equation 2 is prohibitively slow.

An alternative approach is to choose parameter values via maximum likelihood. This can be done quickly, and allows us to make speedy predictions; however, in this case we suffer from problems of data scarcity, particularly early in the tracking process when few observations have been made. The length scale parameters are particularly



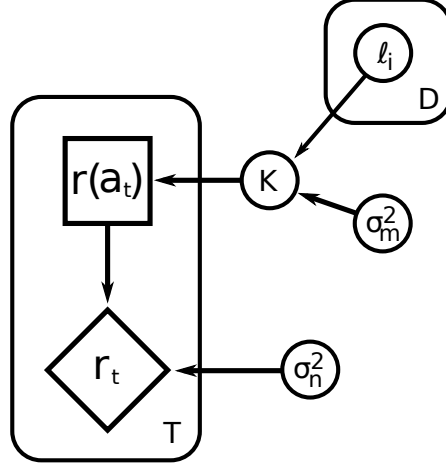


Figure 6: *Graphical model for Bayesian optimization. The  $\ell_i$  are length scales in each dimension,  $\sigma_m^2$  is the magnitude parameter and  $\sigma_n^2$  is the noise level. In our model  $\sigma_m^2$  and  $\sigma_n^2$  follow a uniform prior and the  $\ell_i$  follow independent Student-t priors.*

prone to receiving very poor estimates when there is little data available.

We have found that using informative priors for the length scale parameters and making MAP, rather than ML, estimates at each time step provides a solution to the problems described above. MAP estimates can be made quickly using gradient optimization methods (Rasmussen & Williams, 2006), and informative priors provide resistance to the problems encountered with ML. The experiments in Section 6 place uniform priors on the magnitude and noise parameters and place independent Student-t priors on each length scale parameter. The experiments also use an initial data collection phase of 10 time steps before any adjustment of the parameters is made.

## 5 Algorithm

Since the belief state cannot be computed analytically, we will adopt particle filtering to approximate it. The full algorithm is shown in Algorithm 3.

We refer readers to Doucet et al. (2001) for a more in depth treatment of these sequential Monte Carlo methods. Assume that at time  $t - 1$  we have  $N \gg 1$  particles (samples)  $\{\mathbf{x}_{0:t-1}^{(i)}\}_{i=1}^N$  distributed according to  $p(d\mathbf{x}_{0:t-1}|\mathbf{h}_{1:t-1}, \mathbf{a}_{1:t-1})$ . We can

---

**Algorithm 3** Particle filtering algorithm with gaze control. The algorithm shown here is for partial information strategies. For full information strategies the importance sampling step is done independently for each possible action and the gaze control step is able to use reward information from each possible action to create the new strategy  $\pi_{t+1}(\cdot)$ .

---

### 1. Initialization

**for**  $i = 1$  **to**  $N$  **do**

$$\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$$

Initialize the policy  $\pi_1(\cdot)$  // How this is done depends on the control strategy

**for**  $t = 1 \dots$  **do**

### 2. Importance sampling

**for**  $i = 1$  **to**  $N$  **do**

// Predict the next state

$$\tilde{\mathbf{x}}_t^{(i)} \sim q_t \left( d\mathbf{x}_t^{(i)} \middle| \tilde{\mathbf{x}}_{0:t-1}^{(i)}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t} \right)$$

$$\tilde{\mathbf{x}}_{0:t}^{(i)} \leftarrow \left( \mathbf{x}_{0:t-1}^{(i)}, \tilde{\mathbf{x}}_t^{(i)} \right)$$

$k^* \sim \pi_t(\cdot)$  // Select an action according to the policy

**for**  $i = 1$  **to**  $N$  **do**

// Evaluate the importance weights

$$\tilde{w}_t^{(i)} \leftarrow \frac{p \left( \mathbf{h}_t | \tilde{\mathbf{x}}_t^{(i)}, \mathbf{a}_t = k^* \right) p \left( \tilde{\mathbf{x}}_t^{(i)} | \tilde{\mathbf{x}}_{0:t-1}^{(i)}, \mathbf{a}_{t-1} \right)}{q_t \left( \tilde{\mathbf{x}}_t^{(i)} \middle| \tilde{\mathbf{x}}_{0:t-1}^{(i)}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t} \right)}$$

**for**  $i = 1$  **to**  $N$  **do**

// Normalize the importance weights

$$w_t^{(i)} \leftarrow \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^N \tilde{w}_t^{(j)}}$$

### 3. Gaze control

$$r_t = \sum_{i=1}^N (w_t^{(i)})^2$$

// Receive reward for the chosen action

Incorporate  $r_t$  into the policy to create  $\pi_{t+1}(\cdot)$

### 4. Selection

Resample with replacement  $N$  particles  $\left( \mathbf{x}_{0:t}^{(i)}; i = 1, \dots, N \right)$  from the set  $\left( \tilde{\mathbf{x}}_{0:t}^{(i)}; i = 1, \dots, N \right)$  according to the normalized importance weights  $w_t^{(i)}$

---

approximate this belief state with the following empirical distribution

$$\hat{p}(d\mathbf{x}_{0:t-1}|\mathbf{h}_{1:t-1}, \mathbf{a}_{1:t-1}) \triangleq \frac{1}{N} \sum_{i=1}^N \delta_{\mathbf{x}_{0:t-1}^{(i)}}(d\mathbf{x}_{0:t-1}) \quad .$$

Particle filters combine sequential importance sampling with a selection scheme designed to obtain  $N$  new particles  $\{\mathbf{x}_{0:t}^{(i)}\}_{i=1}^N$  distributed approximately according to  $p(d\mathbf{x}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})$ .

## 5.1 Importance sampling step

The joint distributions  $p(d\mathbf{x}_{0:t-1}|\mathbf{h}_{1:t-1}, \mathbf{a}_{1:t-1})$  and  $p(d\mathbf{x}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})$  are of different dimension. We first modify and extend the current paths  $\mathbf{x}_{0:t-1}^{(i)}$  to obtain new paths  $\tilde{\mathbf{x}}_{0:t}^{(i)}$  using a proposal kernel  $q_t(d\tilde{\mathbf{x}}_{0:t}|\mathbf{x}_{0:t-1}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t})$ . As our goal is to design a sequential procedure, we set

$$q_t(d\tilde{\mathbf{x}}_{0:t}|\mathbf{x}_{0:t-1}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t}) = \delta_{\mathbf{x}_{0:t-1}}(d\tilde{\mathbf{x}}_{0:t-1}) q_t(d\tilde{\mathbf{x}}_t|\tilde{\mathbf{x}}_{0:t-1}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t}) \quad ,$$

that is  $\tilde{\mathbf{x}}_{0:t} = (\mathbf{x}_{0:t-1}, \tilde{\mathbf{x}}_t)$ . The aim of this kernel is to obtain new paths whose distribution

$$q_t(d\tilde{\mathbf{x}}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t}) = p(d\tilde{\mathbf{x}}_{0:t-1}|\mathbf{h}_{1:t-1}, \mathbf{a}_{1:t-1}) q_t(d\tilde{\mathbf{x}}_t|\tilde{\mathbf{x}}_{0:t-1}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t}) \quad ,$$

is as “close” as possible to  $p(d\tilde{\mathbf{x}}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})$ . Since we cannot choose  $q_t(d\tilde{\mathbf{x}}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t}) = p(d\tilde{\mathbf{x}}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})$  because this is the quantity we are trying to approximate in the first place, it is necessary to weight the new particles so as to obtain consistent estimates.

We perform this “correction” with importance sampling, using the weights

$$\tilde{w}_t = \tilde{w}_{t-1} \frac{p(\mathbf{h}_t|\tilde{\mathbf{x}}_t, \mathbf{a}_t) p(d\tilde{\mathbf{x}}_t|\tilde{\mathbf{x}}_{0:t-1}, \mathbf{a}_{t-1})}{q_t(d\tilde{\mathbf{x}}_t|\tilde{\mathbf{x}}_{0:t-1}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t})}.$$

The choice of the transition prior as proposal distribution is by far the most common one. In this case, the importance weights reduce to the expression for the likelihood. However, it is possible to construct better proposal distributions, which make use of more recent observations, using object detectors (Okuma et al., 2004), saliency maps (Itti et al., 1998), optical flow, and approximate filtering methods such as the unscented particle filter. One could also easily incorporate strategies to manage data association and other tracking related issues. After normalizing the weights,  $w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^N \tilde{w}_t^{(j)}}$ , we obtain the following estimate of the filtering distribution:

$$\tilde{p}(d\mathbf{x}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t}) = \sum_{i=1}^N w_t^{(i)} \delta_{\tilde{\mathbf{x}}_{0:t}^{(i)}}(d\mathbf{x}_{0:t}) \quad .$$

Finally a selection step is used to obtain an “unweighted” approximate empirical distribution  $\hat{p}(d\mathbf{x}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})$  of the weighted measure  $\tilde{p}(d\mathbf{x}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})$ . The basic idea is to discard samples with small weights and multiply those with large weights. The use of a selection step is key to making the SMC procedure effective; see Doucet et al. (2001) for details on how to implement this black box routine.

## 6 Experiments

### 6.1 Full Information Policies

In this section, three experiments are carried out to evaluate quantitatively and qualitatively the proposed approach. The first experiment provides comparisons to other control policies on a synthetic dataset. The second experiment, on a similar synthetic dataset, demonstrates how the approach can handle large variations in scale, occlusion and multiple targets. The final experiment is a demonstration of tracking and classification performance on several real videos. For the synthetic digit videos, we trained the first-layer RBMs on the foveated images, while for the real videos we trained factored-RBMs on foveated natural image patches (Ranzato & Hinton, 2010).

The first experiment uses 10 video sequences (one for each digit) built from the MNIST dataset. Each sequence contains a moving digit and static digits in the background (to create distractions). The objective is to track and recognize the moving digit; see Figure 7. The gaze template had  $K = 9$  gaze positions, chosen so that gaze G5 was at the center. The location of the template was initialized with optical flow.

We compare the Hedge learning algorithm against algorithms with deterministic and random policies. The deterministic policy chooses each gaze in sequence and in a particular pre-specified order, whereas the random policy selects a gaze uniformly at random. We adopted the Bhattacharyya distance in the specification of the observation model. A multi-fixation RBM was trained to map the first layer hidden units of three time consecutive time steps into a second hidden layer, and trained a logistic regressor to further map to the 10 digit classes. We used the transition prior as proposal for the particle filter.

Tables 6.1 and 6.1 report the comparison results. Tracking accuracy was measured

	0	1	2	3	4	5	6	7	8	9	AVG.
LEARNED POLICY	<b>1.2</b> (1.2)	<b>3.0</b> (2.0)	<b>2.9</b> (1.0)	<b>2.2</b> (0.7)	<b>1.0</b> (1.9)	<b>1.8</b> (1.9)	<b>3.8</b> (1.0)	<b>3.8</b> (1.5)	<b>1.5</b> (1.7)	<b>3.8</b> (2.8)	<b>2.5</b> (1.6)
DETERMINISTIC POLICY	18.2 (29.6)	536.9 (395.6)	104.4 (69.7)	2.9 (2.2)	201.3 (113.4)	4.6 (4.0)	5.6 (3.1)	64.4 (45.3)	142.0 (198.8)	144.6 (157.7)	122.5 (101.9)
RANDOM POLICY	41.5 (54.0)	410.7 (329.4)	3.2 (2.0)	3.3 (2.4)	42.8 (60.9)	6.5 (9.6)	5.7 (3.2)	80.7 (48.6)	38.9 (50.6)	225.2 (241.6)	85.9 (80.2)

Table 1: *Tracking error (in pixels) on several video sequences using different policies for gaze selection.*

	0	1	2	3	4	5	6	7	8	9	AVG.
LEARNED POLICY	95.62%	<b>100.00%</b>	<b>99.66%</b>	99.33%	<b>99.66%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>98.32%</b>	<b>97.98%</b>	<b>89.56%</b>	<b>98.01%</b>
DETERMINISTIC POLICY	<b>99.33%</b>	<b>100.00%</b>	98.99%	94.95%	5.39%	98.32%	0.00%	29.63%	52.19%	0.00%	57.88%
RANDOM POLICY	98.32%	<b>100.00%</b>	96.30%	<b>99.66%</b>	29.97%	96.30%	89.56%	22.90%	12.79%	13.80%	65.96%

Table 2: *Classification accuracy on several video sequences using different policies for gaze selection.*

in terms of the mean and standard deviation (in brackets) over time of the distance between the target ground truth and the estimate; measured in pixels. The analysis highlights that the error of the learned policy is always below the error of the other policies. In most of the experiments, the tracker fails when an occlusion occurs for the deterministic and the random policies, while the learned policy is successful. This is very clear in the videos at: <http://www.youtube.com/user/anonymousTrack>

The loss of track for the simple policies is mirrored by the high variance results in Table 6.1 (experiments 0, 1, 4, and so on). The average mean and standard deviations (last column of Table 6.1) make it clear that the proposed strategy for learning a gaze policy can be of enormous benefit. The improvements in tracking performance are mirrored by improvements in classification performance (Table 6.1).

Figure 7 provides further anecdotal evidence for the policy learning algorithm. The top sequence shows the target and the particle filter estimate of its location over time. The middle sequence illustrates how the policy changes over time. In particular, it demonstrates that hedge can effectively learn where to look in order to improve tracking performance (we chose this simple example as in this case it is obvious that the center of the eight (G5) is the most reliable gaze action). The classification results over time are shown in the third row.

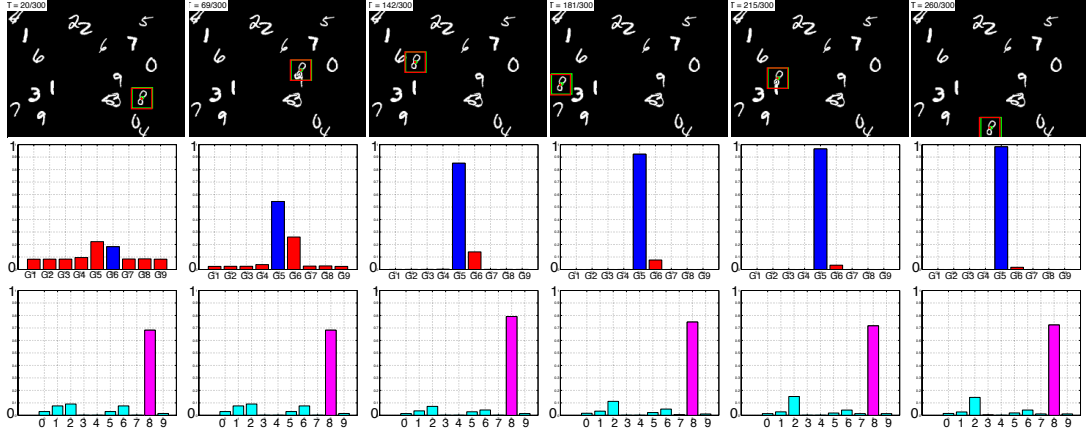


Figure 7: *Tracking and classification accuracy results with the learned policy. **First row:** position of the target and estimate over time. **Second row:** policy distribution over the 9 gazes; hedge clearly converges to the most reasonable policy. **Third row:** cumulative class distribution for recognition.*

The second experiment addresses a similar video sequence, but tracking multiple targets. The image scale of each target changes significantly over time, so the algorithm has to be invariant with respect to these scale transformations. In this case, we used a mixture proposal distribution consisting of motion detectors and the transition prior. We also tested a saliency proposal but found it to be less effective than the motion detectors for this dataset. Figure 8 (top) shows some of the video frames and tracks. The videos allow one to better appreciate the performance of the multi-target tracking algorithm in the presence of occlusions.

Tracking and classification results for the real videos are shown in Figure 8 and the accompanying videos.

## 6.2 Partial Information Policies

In this section, two experiments are carried out to evaluate the performance of the different gaze selection policies.

In the first experiment we compare the performance of each gaze selection method on a data set of several videos of digits from the MNIST data set moving on a black background. The target in each video encounters one or more partial occlusions which the tracking algorithm must handle gracefully. Additionally, each video sequence has

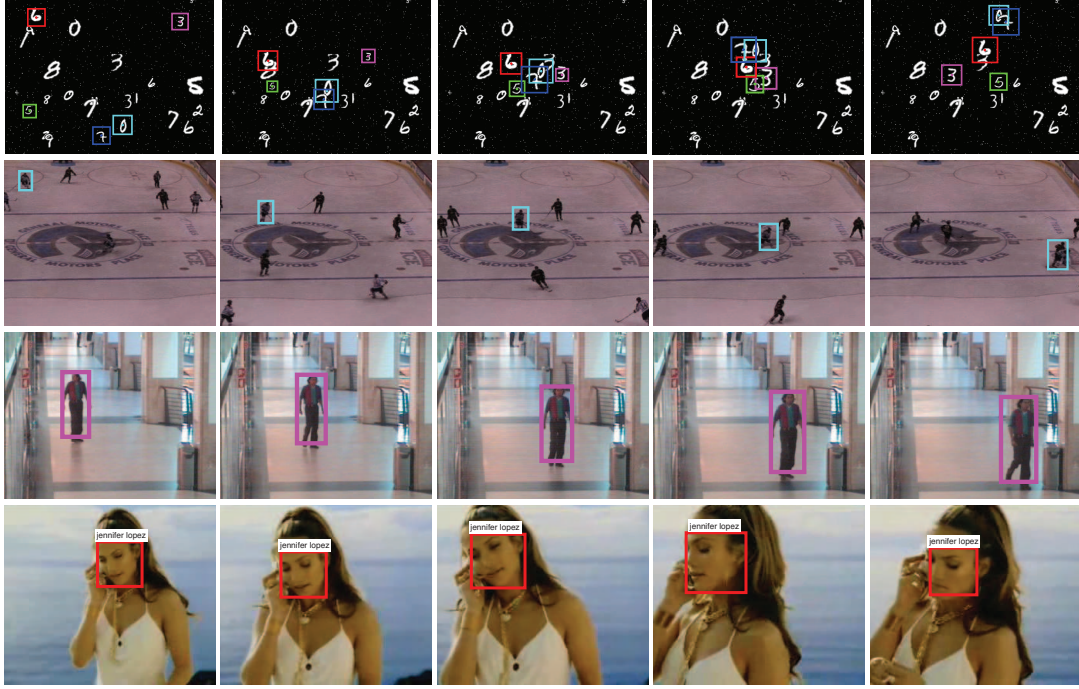


Figure 8: **Top:** Multi-target tracking with occlusions and changes in scale on a synthetic video. **Middle and bottom:** Tracking in real video sequences.

	0	1	2	3	4	5	6	7	8	9	Avg
<b>Bayesopt</b>	5.36 (2.32)	7.92 (2.52)	2.62 (3.89)	4.05 (1.67)	1.70 (5.10)	8.31 (3.35)	4.94 (2.28)	12.09 (3.53)	1.52 (2.76)	9.06 (1.66)	<b>5.76</b> <b>(2.91)</b>
<b>Hedge</b>	2.97 (1.56)	3.20 (2.19)	2.97 (1.99)	2.92 (2.00)	3.14 (1.80)	2.96 (2.08)	2.86 (1.96)	2.98 (1.76)	2.81 (1.64)	3.15 (3.73)	<b>3.00</b> <b>(2.07)</b>
<b>EXP3</b>	3.18 (5.05)	3.03 (10.08)	65.46 (3212.16)	91.81 (3671.66)	2.62 (2.35)	7.20 (303.29)	67.54 (2346.82)	2.97 (3.99)	3.06 (2.71)	77.01 (3135.17)	<b>32.39</b> <b>(1269.33)</b>

Table 3: Tracking error on several video sequences using different methods for gaze selection. The table shows mean tracking error as well as the error variance (in brackets) over a single test sequence.

been corrupted with 30% noise. We measure the error between the estimated track and the ground truth for each gaze selection method, and demonstrate that Bayesian optimization preforms comparably to Hedge, but that EXP3 is not able to reach a satisfactory level of performance. We also demonstrate qualitatively that the Bayesian optimization approach learns good gaze selection policies on this data set.

Our second experiment provides evidence that the Bayesian optimization method can generalize to real world data.

Table 3 reports the results from our first experiment. The table shows the mean tracking error, measured by averaging distance between the estimated and ground truth

track over the entire video sequence. Here we see that the Bayesian optimization approach compares favorably to Hedge in terms of tracking performance, and that EXP3 preforms substantially worse than the other two methods. Although Hedge preforms marginally better than Bayesian optimization, it is important to remember that Bayesian optimization solves a significantly more difficult problem. Hedge relies on discretizing the action space, and must have access to the rewards for all possible actions at each time step. In contrast, Bayesian optimization considers a fully continuous action space, and receives reward information only for the chosen actions.

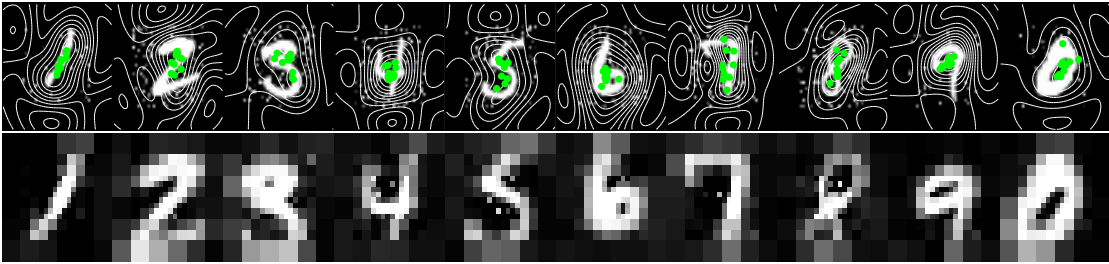


Figure 9: **Top:** Digit templates with the estimated reward surfaces superimposed. Markers indicate the best fixation point found in each of ten runs. **Bottom:** A visualization of the image found by averaging the best fixation points found across ten runs.

Figure 9 shows the reward surfaces learned for each digit by Bayesian optimization, as well as a visualization of the overall best fixation points using data aggregated across ten runs. The optimal fixation points found by the algorithm are tightly clustered, and the resulting observations are very distinguishable.

In our second experiment we use the Youtube celebrity dataset from Kim et al. (2008). This data set consists of several videos of celebrities taken from Youtube and is challenging for tracking algorithms as the videos exhibit a wide variety of illuminations, expressions and face orientations. We run our tracking model using Bayesian optimization to learn a gaze selection policy on this data set, and present some results in Figure 10. Although we report only qualitative results from this experiment, it provides anecdotal evidence that Bayesian optimization is able to form a good gaze selection policy on real world data.



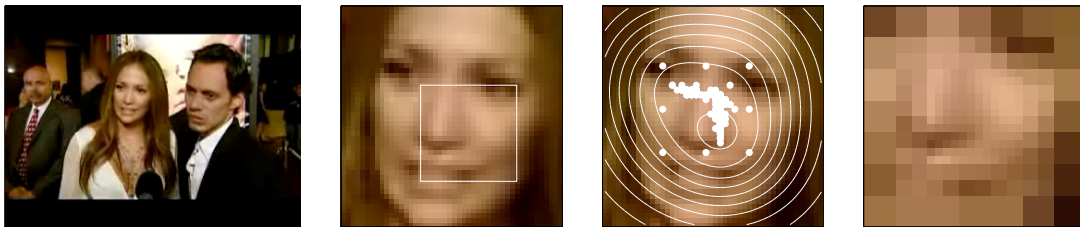


Figure 10: *Results on a real data set. **Far left:** An example frame from the video sequence. **Center left:** The tracking template with the optimal fixation window highlighted. **Center right:** The reward surface produced by Bayesian optimization. The white markers show the centers of each fixation point in a single tracking run. **Right:** Input to the observation model when fixating on the best point. (Best viewed from a distance).*

## 7 Conclusions and Future Work

We have proposed a decision-theoretic probabilistic graphical model for joint classification, tracking and planning. The experiments demonstrate the significant potential of this approach. We examined several different strategies for gaze control in both the full and partial information settings. We saw that a straightforward generalization of the full information policy to partial information gave poor performance and we proposed an alternative method which is able not only to perform well in the presence of partial information but also allows us to expand the set of possible fixation points to be a continuous domain.

There are many routes for further exploration. In this work we pre-trained the (factored)-RBMs. However, existing particle filtering and stochastic optimization algorithms could be used to train the RBMs online. Following the same methodology, we should also be able to adapt and improve the target templates and proposal distributions over time. This is essential to extend the results to long video sequences where the object undergoes significant transformations (e.g. as is done in the predator tracking system (Kalal et al., 2010)).

Deployment to more complex video sequences will require more careful and thoughtful design of the proposal distributions, transition distributions, control algorithms, template models, data-association and motion analysis modules. Fortunately, many of the solutions to these problems have already been engineered in the computer vision, tracking and online learning communities. Admittedly, much work remains to be done.

Saliency maps are ubiquitous in visual attention studies. Here, we simply used standard saliency tools and motion flow in the construction of the proposal distributions for particle filtering. There might be better ways to exploit the saliency maps, as neurophysiological experiments seem to suggest (Gottlieb et al., 1998).

One of the most interesting avenues for future work is the construction of more abstract attentional strategies. In this work, we focused on attending to regions of the visual field, but clearly one could attend to subsets of receptive fields or objects in the deep appearance model.

The current model has no ability to recover from a tracking failure. It may be possible to use information from the identity pathway (i.e. the classifier output) to detect and recover from tracking failure.

A closer examination of the exploration/exploitation tradeoff in the tracking setting is in order. For instance, the methods we considered assume that future rewards are independent of past actions. This assumption is clearly not true in our setting, since choosing a long sequence of very poor fixation points can lead to tracking failure. We can potentially solve this problem by incorporating the current tracking confidence into the gaze selection strategy. This would allow the exploration/exploitation trade off to be explicitly modulated by the needs of the tracker, e.g. after choosing a poor fixation point the selection policy could be adjusted temporarily to place extra emphasis on exploiting good fixation points until confidence in the target location has been recovered. Contextual bandits provide a framework for integrating and reasoning about this type of side-information in a principled manner.

## Acknowledgments

We thank Ben Marlin, Kenji Okuma, Marc’Aurelio Ranzato and Kevin Swersky. This work was supported by CIFAR’s NCAP program and NSERC.

## References

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R.E. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *focs*, pp. 322. Published by the IEEE

- Computer Society, 1998a.
- Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R.E. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2001. ISSN 0097-5397.
- Auer, Peter, Cesa-Bianchi, Nicolò, Freund, Yoav, and Schapire, Robert E. Gambling in a rigged casino: the adversarial multi-armed bandit problem. Technical Report NC2-TR-1998-025, 1998b.
- Bazzani, L., de Freitas, N., and Ting, J.A. Learning attentional mechanisms for simultaneous object tracking and recognition with deep networks. *NIPS 2010 Deep Learning and Unsupervised Feature Learning Workshop*, 2010.
- Brochu, E., Cora, V.M., and de Freitas, N. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. Technical report, University of British Columbia, 2010.
- Colombo, John. The development of visual attention in infancy. *Annual Review of Psychology*, pp. 337–367, 2001.
- Doucet, A., de Freitas, N., and Gordon, N. Introduction to sequential Monte Carlo methods. In Doucet, A., de Freitas, N., and Gordon, N J (eds.), *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.
- Freund, Yoav and Schapire, Robert E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.
- Gaborski, R., Vaingankar, V., Chaoji, V., Teredesai, A., and Tentler, A. Detection of inconsistent regions in video streams. In *Proc. SPIE Human Vision and Electronic Imaging*. Citeseer, 2004.
- Gao, D., Mahadevan, V., and Vasconcelos, N. The discriminant center-surround hypothesis for bottom-up saliency. *Advances in neural information processing systems*, 20, 2007.

- Girard, B. and Berthoz, A. From brainstem to cortex: Computational models of saccade generation circuitry. *Progress in Neurobiology*, 77(4):215 – 251, 2005.
- Gottlieb, Jacqueline P., Kusunoki, Makoto, and Goldberg, Michael E. The representation of visual salience in monkey parietal cortex. *Nature*, 391:481–484, 1998.
- Hinton, GE and Salakhutdinov, RR. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.
- Isard, M and Blake, A. Contour tracking by stochastic propagation of conditional density. In *European Computer Vision Conference*, pp. 343–356, 1996.
- Itti, L., Koch, C., and Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254 –1259, 1998.
- Jones, D.R., Perttunen, C.D., and Stuckman, B.E. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993. ISSN 0022-3239.
- Kalal, Z., Mikolajczyk, K., and Matas, J. Face-tld: Tracking-learning-detection applied to faces. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pp. 3789–3792. IEEE, 2010.
- Kavukcuoglu, K., Ranzato, M.A., Fergus, R., and Le-Cun, Yann. Learning invariant features through topographic filter maps. In *Computer Vision and Pattern Recognition*, pp. 1605–1612, 2009.
- Kim, M., Kumar, S., Pavlovic, V., and Rowley, H. Face tracking and recognition with visual constraints in real-world videos. *IEEE Conf. Computer Vision and Pattern Recognition*, 2008.
- Koch, C. and Ullman, S. Shifts in selective visual attention: towards the underlying neural circuitry. *Hum Neurobiol*, 4(4):219–27, 1985.
- Köster, Urs and Hyvärinen, Aapo. A two-layer ICA-like model estimated by score matching. In *International Conference of Artificial Neural Networks*, pp. 798–807, 2007.

- Larochelle, Hugo and Hinton, Geoffrey. Learning to combine foveal glimpses with a third-order Boltzmann machine. In *Neural Information Processing Systems*, 2010.
- Lee, H., Grosse, R., Ranganath, R., and Ng, A.Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *International Conference on Machine Learning*, 2009.
- McNaughton, Bruce L., Battaglia, Francesco P., Jensen, Ole, Moser, Edvard I., and Moser, May-Britt. Path integration and the neural basis of the 'cognitive map'. *Nature Reviews Neuroscience*, 7(8):663–678, 2006.
- Okuma, Kenji, Taleghani, Ali, de Freitas, Nando, and Lowe, David G. A boosted particle filter: Multitarget detection and tracking. In *ECCV*, 2004.
- Olshausen, B. A. and Field, D. J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.
- Olshausen, B.A., Anderson, C.H., and Van Essen, D.C. A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *The Journal of Neuroscience*, 13(11):4700, 1993a. ISSN 0270-6474.
- Olshausen, Bruno A., Anderson, Charles H., and Essen, David C. Van. A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *Journal of Neuroscience*, 13:4700–4719, 1993b.
- Postma, Eric O., van den Herik, H. Jaap, and Hudson, Patrick T. W. SCAN: A scalable model of attentional selection. *Neural Networks*, 10(6):993 – 1015, 1997.
- Ranzato, M.A. and Hinton, G.E. Modeling pixel means and covariances using factorized third-order Boltzmann machines. In *Computer Vision and Pattern Recognition*, pp. 2551–2558, 2010.
- Rasmussen, C.E. and Williams, C.K.I. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006. ISBN 9780262182539. URL <http://books.google.ca/books?id=vWtwQgAACAAJ>.
- Rensink, Ronald A. The dynamic representation of scenes. *Visual Cognition*, pp. 17–42, 2000.

- Rosa, M.G.P. Visual maps in the adult primate cerebral cortex: Some implications for brain development and evolution. *Brazilian Journal of Medical and Biological Research*, 35:1485 – 1498, 2002.
- Srinivas, N., Krause, A., Kakade, S.M., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. *International Conference on Machine Learning*, 2010.
- Swersky, K., Chen, Bo, Marlin, B., and de Freitas, N. A tutorial on stochastic approximation algorithms for training restricted Boltzmann machines and deep belief nets. In *ITA Workshop*, pp. 1–10, 2010.
- Swersky, K., Buchman, D., Marlin, B.M., and de Freitas, N. On autoencoders and score matching for energy based models. *International Conference in Machine Learning*, 2011.
- Taylor, G.W., Sigal, L., Fleet, D.J., and Hinton, G.E. Dynamical binary latent variable models for 3D human pose tracking. In *Computer Vision and Pattern Recognition*, pp. 631–638, 2010.
- Torralba, A., Oliva, A., Castelhana, M.S., and Henderson, J.M. Contextual guidance of eye movements and attention in real-world scenes: The role of global features in object search. *Psychological review*, 113(4):766, 2006.
- Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.A. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning*, pp. 1096–1103, 2008.
- Welling, M., Rosen-Zvi, M., and Hinton, G. Exponential family harmoniums with an application to information retrieval. *Neural Information Processing Systems*, 17: 1481–1488, 2005.
- Zhang, L., Tong, M.H., and Cottrell, G.W. Sunday: Saliency using natural statistics for dynamic analysis of scenes. In *Proceedings of the 31st Annual Cognitive Science Conference, Amsterdam, Netherlands*. Citeseer, 2009.